SDN 환경에서 오토 인코더 모델을 활용한 플로우 분류 기법

전장군*, 김희찬**, 이상민**, 김남기***
*경기대학교 전자공학과, **, ***경기대학교 컴퓨터공학부

201812285@kyonggi.ac.kr, train1212@kyonggi.ac.kr, d9249@kyonggi.ac.kr, ngkim@kyonggi.ac.kr

A Study on the Flow Classification Method Using Auto-Encoder Model in SDN Environment

Jeon Janggun*, Kim Heechan**, Lee Sangmin**, Kim Namgi***
*Dept. of Electric Engineering Kyonggi Univ.,
**, ***Dept. of Computer Engineering Kyonggi Univ.

요 약

본 논문에서는 플로우 분류 기법들 중 포트 기반 분류, 페이로드 기반 분류, 머신러닝 기반 분류에 대해 간단히 소개한다. 그리고 최근 각광받고 있는 기술인 딥러닝을 활용해 플로우를 분류하는데 딥러닝 모델 중 오토 인코더 모델인 VAE 와 VQ-VAE 을 활용하여 좀 더 정확히 플로우를 분류할 수 있는 기법을 제안한다.

Ⅰ. 서론

최근 IT기술의 발전 및 모바일 기기의 보급 확대로 인해 언제 어디서든 인터넷에 연결할 수 있음에 따라 네트워크 트래픽이 폭발적으로 증가하고 있다. 이에 따라 네트워크 트래픽을 보다 더 효율적으로 관리하기 위해 네트워크에 대한 연구가 활발히 진행되어 왔으며, 그 중 Software Defined Network(SDN)[1]이 많이 연구되고 있다. SDN에서 네트워크 애플리케이션마다 데이터 손실, 대역폭, 시간 민감성 등 필요한 요구사항들이 다르기 때문에 플로우의 특성을 파악해 플로우를 분류하는 것이 중요하다. 플로우 분류를 잘 하게 되면 최적의 라우팅 경로를 설정할 수 있게 된다. 본 논문에서는 비지도 학습 모델 중 하나인 오토 인코더 모델을 활용해 SDN환경에서 플로우를 분류하는 기법을 제안한다.

Ⅱ. 관련 연구

플로우의 특성을 이용해 플로우를 분류하는 연구는 그 이전부터 활발히 수행되고 있다. 그 중 포트 기반 분류, 페이로드 기반 분류, 머신러닝 기반 분류 기법에 대해 간단히 소개한다. 포트 기반 분류[2]는 과거에 많이 사용되던 분류기법으로, 고정된 포트 번호를 이용하여 분류하는 방법이다. 그러나, 이 방법은 고정 포트 번호를 사용하지 않는 서비스의 등장으로 인해 분류하기 어렵다는 문제점이 존재한다. 페이로드 기반 분류[3]는 현재 가장 많이 사용되는 분류 기법으로, 패킷의 내부를 분석해 분류하는 방법이다. 하지만, 높은 메모리 사용량, 전문가 의존적, 패킷이 암호화 되었을 경우 분류하기 어렵다는 문제점이 존재한다. 머신러닝 기반 분류[4]는 머신러닝 알고리즘을 사용한 분류 기법으로, 앞서 소개한 분류 기법들의 문제점들을 모두 해결한다. 특히 최근 각광받고 있는 기술인 딥러닝을 이용하여 분류하는 기법들이 많이 연구되고 있다.

Ⅲ. 본론

관련 연구에서 플로우를 분류하는 다양한 기법들에 대해 소개하였다. [5]에서 딥러닝을 활용한 플로우 분류 기법에 대한 연구가 수행되었으나, 본 논문에서는 후속 연구로 Vector Quantised-Variational AutoEncoder (VQ-VAE) 모델 [6]을 추가로 이용한다.

다음 그림 1 은 본 논문에서 제안하는 플로우 분류 기법 모델이다.

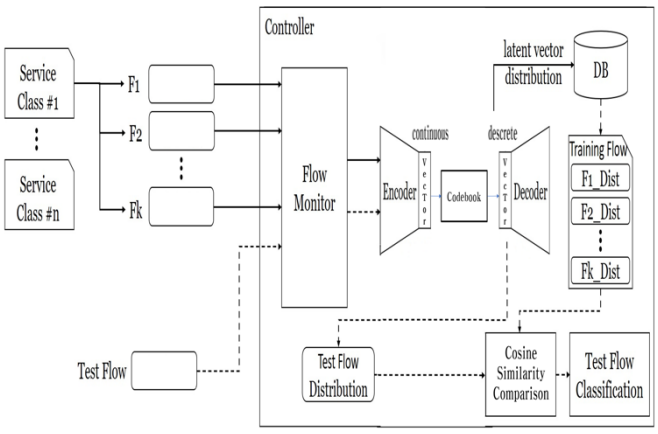

그림 1. 제안하는 플로우 분류 모델

실험과정은 크게 data preprocessing, model creating, model training, test flow feature extraction, flow classification로 나뉜다.

Data preprocessing은 입력데이터인 flow data을 VQ-VAE모델에 적합하게 변환하는 과정으로, flow data는 flow statistics와 service class label로 나뉘어지는데, flow statistics는 data scale을 standardize해서 x로, service class label은 ont-hot encoding을 해서 y로 사용한다.

Model creating은 모델의 구조를 생성하고, 초기화하며 손실함수를 정의하는 과정이다. 기존의 논문[5]과 동일한 fully-connected 2-layer encoder, decoder network을 생성하고, 그 사이에 vector quantizer layer을 추가해서 VQ-VAE을 설계한다. VQ layer은 인코더 출력 latent vector을 동일한 크기로 양자화할 vector들의 테이블(codebook)을 내부 parameter로 저장한다. Codebook의 size는 hyperparameter로 정하고, codebook의 weight의 초기값은 standard uniform or normal distribution로 설정한다.

본 논문에서 설계한 VQ-VAE network는 input data인 x을 encoder에 입력하면 continuous latent vector을 출력하는데, 이것을 VQ layer에 입력하면, codebook의 vector들과 유클리드 거리를 비교해서 가장 유사한 vector로 대체되어 출력된다. 이때, argmin operation이 사용되는데 이 operation은 미분 불가능한 연산이기 때문에, VQ layer와 encoder출력 latent vector 간의 reconstruction loss을 구할 때는, VQ-layer의 목적이 continuous latent vector을 quantization하는 것이기 때문에 encoder출력을 codebook에 맞추도록 codebook loss을 구한다. 그런데 codebook은 reconstruction loss가 skip되어 전달되지 않기 때문에 양자화 벡터 공간이 무한하게 다양한 값을 가질 수 있어, codebook vector들이 encoder 출력과 유사한 discrete한 값을 갖도록 commitment loss을 구한다. 이때 이 값은 codebook loss보다는 작은 값이어야 한다. 그 다음 VQ layer가 discrete latent vector을 decoder에 전달되며, 모델의 output인 reconstruction x와 함께 추가적으로 codebook&commitment loss, latent vector을 함께 출력하는 구조를 VQ-VAE가 가지게 된다.

Model training단계에서는, 설정된 epoch만큼 train x를 VQ-VAE network에 forwarding해서 출력된 loss들과 출력된 reconstruction x를 통해 구한 reconstruction loss을 backwarding하는 학습을 반복한다. 학습이 완료되면 마지막 epoch에서 얻어진 train x에 대한 latent vector들을 service class별로 k개 씩 추출해서 buffer에 저장한다.

Test flow feature extraction단계에서는, model의 weight들을 고정시켜놓고, test x 에 대한 latent vector들을 모두 추출한다.

Flow classification은 test latent vector와 buffer에 저장된 vector들의 유사도를 비교하는 과정이다. Test lantent vector와 가장 유사도가 높은 vector가 속한 train y을 보고 test flow을 그 train y가 나타내는 클래스로 추정한다. 그리고 test y와 비교해서 정확하게 분류했는지 판단해, 전체의 정확도를 출력한다.

Ⅳ. 결론

본 논문에서는 SDN 환경에서 딥러닝 모델 기법 중 하나인 오토 인코더 모델을 활용한 플로우 분류 기법을 제안하였다. 기존 분류 방법들은 앞서 관련 연구에서 소개하였듯 여러 단점들이 많으나 머신러닝 분류 기법은 이러한 단점들을 모두 해결하고 정확도도 높아 추후 많은 연구가 진행될 것으로 예상된다. 특히 본 논문에서 제안한 플로우 분류 기법은 최신 딥러닝 모델을 활용하므로 기존 분류 기법보다 좀 더 높은 정확도로 분류할 수 있을 것으로 예상된다. 추후 본 논문에서 제안한 모델을 학습 후 기존 논문과 비교해 더 높은 정확도를 가지는 지 비교할 예정이다.

ACKNOWLEDGMENT

본 논문은 교육부의 재원으로 한국연구재단의 지원 (NRF-2020R1A6A1A03040583)과 경기도에서 지원하는 경기도지역협력연구센터(GRRC)사업의 결과로 수행되었음

참 고 문 헌

[1] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey, " in Proceedings of the IEEE, vol. 103, no. 1, pp. 14-76, 2015.

[2] Schneider, P., "TCP / IP Classification Based on Port Numbers", Division of Applied Sciences 2138, 1996.

[3] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network Traffic Classification Using Correlation Information," in IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 1, pp. 104-117, 2013.

[4] Parsaei, Mohammad Reza, Mohammad Javad Sobouti, and Reza Javidan. "Network traffic classification using machine learning techniques over software defined networks." International Journal of Advanced Computer Science and Application 8.7, 2017.

[5] 장예훈, "SDN 환경에서 딥러닝을 활용한 플로우 분류 기법 연구." 국내석사학위논문 경기대학교 대학원, 2021.

[6] Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning.", Advances in neural information processing system 30, 2017.